Using WordNet to Build Lexical Sets for Italian Verbs

Anna Feltracco
Fondazione Bruno Kessler
Università di Pavia, Italy
feltracco@fbk.eu

Lorenzo Gatti
Fondazione Bruno Kessler
Università di Trento, Italy
l.gatti@fbk.eu

Simone Magnolini
Fondazione Bruno Kessler
Università di Brescia, Italy
magnolini@fbk.eu

Bernardo Magnini
Fondazione Bruno Kessler
Povo-Trento, Italy
magnini@fbk.eu

Elisabetta Jezek
Università di Pavia
Pavia, Italy
jezek@unipv.it

Abstract

We present a methodology for building lexical sets for argument slots of Italian verbs. We start from an inventory of semantically typed Italian verb frames and through a mapping to WordNet we automatically annotate the sets of fillers for the argument positions in a corpus of sentences. We evaluate both a baseline algorithm and a syntax driven algorithm and show that the latter performs significantly better in terms of precision.

1 Introduction

In this paper we present a methodology for building lexical sets for argument slots of Italian verbs. Lexical sets (Hanks, 1996) are paradigmatic sets of words which occupy the same argument positions for a verb, as found in a corpus. For example, for the verb *read*, the following set can be built by observing the lexical fillers of the object position in the BNC corpus:

(1) *read* {book, newspaper, bible, article, letter, poem, novel, text, page, passage, ...}

To collect lexical sets for Italian verbs, we use the lexical resource T-PAS (Jezek et al., 2014), an inventory of typed predicate argument structures for Italian manually acquired from corpora through inspection and annotation of actual uses of the analyzed verbs. In the current version of the T-PAS resource, only the verb is tagged in the annotated corpus, while the lexical items for each argument slots are not. Thus, the annotation of the lexical sets will enrich the actual version of the resource and will open to experiments for automatically extending its coverage.

A relevant step in our methodology is the annotation of the lexical items for argument positions in sentences. A previous work (Jezek and Frontini, 2010) has already outlined an annotation scheme for this purpose, and highlighted its benefits for NLP applications. In that work, however, the annotation of lexical sets was intended as manual, whereas the methodology we propose here is conceived for automatic annotation, and exploits an existing external resource. Under this perspective our work is related to semantic role labeling (Palmer et al., 2010).

This paper is organized as follows. Section 2 introduces the T-PAS resource; in Section 3 the lexical set population task is defined, and in Section 4 the experimental setting is presented. Section 5 discusses the results and is followed by the error analysis in Section 6. Finally, Section 7 provides some conclusions and directions for future work.

2 Overview of the T-PAS Resource

T-PAS, Typed Predicate Argument Structures, is a repository of verb patterns acquired from corpora by manual clustering of distributional information about Italian verbs (Jezek et al., 2014).

The resource has been developed following the lexicographic procedure called Corpus Pattern Analysis, CPA (Hanks, 2004). In particular, in the resource T-PASs are semantically motivated and are identified by analysing examples found in a corpus of sentences, i.e. a reduced version of ItWAC (Baroni and Kilgarriff, 2006).

After analyzing a sample of 250 concordances of the verb in the corpus, the lexicographer defines each T-PAS recognising its relevant structure and identifying the Semantic Types (STs) for each argument slots by generalizing over the lexical sets observed in the concordances; as an exam-

2 [[Human]] **divorare** [[Document]]
[[Human]] legge [[Document]] con grande interesse

Figure 1: T-PAS#2 for the verb *divorare*.

Figure 2: Lexical Set identification for T-PAS#2 for the verb *divorare*.

ple, Figure1 shows the T-PAS#2 of the verb *divorare*: [[Human]] divorare [[Document]] (Eng. to devour), where [[Document]] stands for {*libro, romanzo, saggio*} (Eng. {book, newspaper, essay}) (Figure 2). STs are chosen among a list of about 230 corpus-derived semantic classes compiled by applying the CPA procedure to the analysis of concordances for about 1500 English and Italian verbs (Jezek et al., 2014)[1]. If no generalization is possible, the lexical set is listed. Finally, the lexicographer associates the instances in the corpus to the corresponding T-PAS and adds a free-text description of its sense (Figure 1). The T-PAS resource thus lists the analyzed verbs[2], the identified T-PASs for each verb, the annotated instances for the T-PAS in the corpus.

In the next Sections, we will define the lexical set population task and describe the experiment we ran and its evaluation.

3 Task Definition

The aim of our system is to automatically derive lexical sets corresponding to the STs in the T-PAS resource. The task is defined as follows. The system receives as input *(i)* a T-PAS of a certain verb and *(ii)* a sentence associated to that T-PAS in the resource. The system should correctly mark (where present) the lexical items or the multiword expressions correspondent to the STs of each argument position specified by the T-PAS (i.e. sentence annotation step). By replicating this annotation for all the sentences of a T-PAS, the system will build the lexical set for a specific ST in a specific T-PAS (i.e. lexical set population step).

[1]Labels for STs in T-PAS are in English, as in the corresponding English resource PDEV (Hanks and Pustejovsky, 2005).

[2]The current version of T-PAS contains 1000 analyzed average polysemy verbs, selected on the basis of random extraction of 1000 lemmas out of the total set of fundamental lemmas of Sabatini Coletti (2007).

For instance, example (2) shows the T-PAS#1 of the verb *preparare* (Eng. to prepare) and a sentence associated to it.

(2) [[Human]] **preparare** [[Food | Drug]]
"La nonna, prima di infornare le patate, **prepara** una torta"
(Eng. "the grandmother, before baking the potatoes, **prepares** a cake")

In this case, the system should identify *nonna* (Eng. grandmother) as a lexical item for [[Human]]-SUBJ and *torta* (Eng. cake) for [[Food]]-OBJ. If this annotation is repeated for all the sentences of the T-PAS#1 of the verb *preparare*, the system will build the lexical set for the ST [[Human]] in Subject position in the T-PAS, such as {*nonna, chef, Gino, bambina, ..*}, and for [[Food]] in object position, such as {*torta, zuppa, pasta, panino, ..*}.

4 Experimental Setting

In order to identify possible candidate items for a ST, the system uses information from MultiWordNet (Pianta et al., 2002)(from now on MWN); e.g. to derive that "grandmother" is a human being and associate it to the ST [[Human]] and that "cake" is a type of food and associate it to the ST [[Food]]. The task, thus, required an initial mapping between the T-PAS resource and MWN. Then, we compared a naive Baseline algorithm and a more elaborated algorithm that we called LEA, Lexical Set Extraction Algorithm. Finally, to evaluate the performance of our methodology we also created a gold standard.

ST to Synset mapping. For our experiment, the list of STs used in the T-PAS resource was automatically mapped onto corresponding WordNet 1.6 synsets. For instance, the ST [[Human]] was mapped to all the synsets for the noun *human* (i.e. *human#n*). Manual inspection was limited to the case in which there is no exact match between a ST and a synset (e.g. by associating "atmospheric-phenomenon" to [[Weather Event]]).

The Baseline algorithm. The Baseline algorithm identifies possible candidate members of the lexical set corresponding to a certain ST for a certain T-PAS by *(i)* lemmatizing each sentence using TextPro (Pianta et al., 2008), *(ii)* checking if each lemma is in MWN and *(iii)* determining whether

the lemma belongs to a synset that was mapped to the ST, or if it is an hyponym of one such synsets.

For instance, in example (2), the Baseline lemmatizes the sentence and selects as possible candidates the nouns of the sentence, i.e. *nonna*, *torta* and *patate*. The Italian lemma *nonna* is thus searched in MWN and the correspondent English lemmas *grandma#n#1, grandmother#n#1, granny#n#1, grannie#n#1* are found. Since none of these synset lemmas match with [[Human]], [[Food]] or [[Drug]], the MWN hierarchy is traversed until *human#n#1* is found, which is mapped to [[Human]]. The same is done for *torta* and *patate*, until [[Food]] is found. Thus, for (2), the Baseline identifies *nonna* as [[Human]] and *torta* and *patate* as [[Food]] (with *patate* being a misclassified item, as it is not referred to the verb *preparare*).

The LEA algorithm. Compared to the Baseline algorithm, the LEA algorithm takes into account also the dependency tree of the sentence, named entities as recognized by TextPro, and multiword expressions.

It starts by *(i)* finding the position of the verb in an example and considering as valid candidate only the chunks that are a subject, direct object or complement of the verb according to the TextPro dependency tree. With respect to the Baseline, this leads to a more precise identification of the items for the argument slots of just the verb we are considering. For instance, in (2) we expect the algorithm to correctly identify *nonna* as [[Human]] and *torta* as [[Food]], but not proposing *patate* (as the Baseline does).

The LEA algorithm also *(ii)* checks if the verb allows the same ST for subject and object, as in the T-PAS#3 of *pettinare*: [[Human1]] pettinare [[Human2]] (Eng. to comb someone's hair). In the sentence "La mamma pettina il bambino" (Eng. The mum combs the baby), LEA will correctly propose *mamma* as [[Human1]] and *baby* as [[Human2]]. In this case, it also checks if the verb is in passive form and swaps the items for subject and object position as needed, improving the precision with respect to the Baseline.

Furthermore, the algorithm *(iii)* checks if the chunk contains/overlaps with proper names related to persons, organizations and locations detected by TextPro, and, if this is the case, checks the corresponding type of named entity against the ST allowed by the T-PAS frame (e.g. *Maria Rossi* → Person → [[Human]]). Since the Baseline recognizes only named entities that are in MWN, we expect this algorithm to identify more items.

Finally, LEA *(iv)* looks for multiword expressions in a chunk by checking if the combination exists in MWN. For instance, in "La nonna prepara la conserva di frutta" (Eng.: the grandmother prepares the fruit conserve), LEA should identify *conserva di frutta* as [[Food]] (while the Baseline identifies only the token *frutta*).

The LEA algorithm, thus, should recognize as valid only the items for a certain argument slot of the analyzed verb (and not for other verbs in the sentence), solve major cases of same ST in different slots and identify named entities and multiword expressions.

Gold Standard. We created a gold standard for the task by manually annotating 500 examples. We asked three annotators to mark the lexical items or the multiword expressions that correspond to the STs, without annotating pronouns or relative clauses. We selected the 500 sentences by extracting 10 sentences for 10 different STs in 5 different T-PASs (for a total of 50 different T-PASs belonging to 47 verbs). In particular, we chose, among all the STs within the [[Inanimate]] hierarchy, 10 types that are used in at least 5 different T-PAS, each of them having at least 10 (potential) sentences associated in the corpus resource. For example, we selected [[Food]] and annotated 10 sentences for T-PAS#1 of *mangiare* "[[Human]] mangiare [[Food]]" (Eng. to eat), since *(i)* there are at least 5 verbs with a T-PAS containing [[Food]], like *mangiare* itself and *(ii)* we have at least 10 sentences available for each of these five T-PASs [3]. This selection of few STs was intended to better compare performances of the algorithms for different lexical sets.

The gold standard annotation resulted in a total of 981 annotated tokens out of 15090 (the average sentence length being 30.18 tokens).

5 Results

For what concerns sentence annotation, we evaluate overall precision, recall and F-measure, con-

[3] This is mainly a selection criteria. Considering that we analyzed a limited number of examples for each verb, and that more than one ST can be specified for each argument slot, it is also possible that none of the sentences extracted for a ST for a verb instantiate that particular ST.

sidering as a positive match when the algorithms agree with the gold standard in recognizing a token as an item (or part of the item in case of multiword expressions) instantiating a ST for a precise position.

Compared to the Baseline, the LEA algorithm registers a significant higher value for precision (see Automatic Mapping in Table 1). This is not surprising, as the Baseline considers as valid all the items in the sentence that can correspond to the ST, without taking into account if they are in the argument position required by the T-PAS or not. On the contrary, the LEA algorithm also considers the syntactic structure, thus lowering the false positives rate; the downside effect is that its recall is lower than the one of the Baseline.

Automatic mapping			
	Precision	Recall	F1
Baseline	0.28	0.42	0.34
LEA	0.70	0.25	0.37
Mapping with manual revision			
Baseline	0.30	0.52	0.38
LEA	0.72	0.32	0.44

Table 1: Results for sentence annotation for the Baseline Algorithm and the LEA Algorithm.

We also measured the similarity between the 5 most populated lexical sets in the gold standard (from 6 to 15 tokens in 10 sentences) and their correspondent lexical sets built by the two algorithms (see Table 2), by calculating the Dice's coefficient[4] (van Rijsbergen, 1979). For example, we compare the lexical set of the T-PAS#1 of *crollare*: [[Building]] crollare (Eng. to fall down) {*e.g. casa, muro, torre*} with the lexical set for the same ST in the same T-PAS derived by the Baseline and LEA.

Results show that both the Baseline and LEA do not reach high overlap. In fact, even if LEA has an high precision in identifying the members of the lexical set, the low recall penalizes the amount of items it can detect given few sentences to annotate. On the contrary, the Baseline is favored by a higher recall, but its low precision causes major differences with the gold standard sets. For these reasons, we believe that on a broader scale, the higher precision for LEA is more advisable with respect to the Baseline.

[4]Dice's coefficient measures how similar two sets are by dividing the number of shared elements of the two sets by the total number of elements they are composed by. This produces a value from 1, when both sets share all elements, to 0, when they have no element in common.

	Baseline	LEA
Cuocere#2-SBJ-[[Food]]	0.54	0.57
Crollare#1-SBJ-[[Building]]	0.40	0.25
Dirottare#1-OBJ-[[Vehicle]]	0.72	0.50
Prescrivere#2-OBJ-[[Drug]]	0.42	0.46
Togliere#4-OBJ-[[Garment]]	0.45	0.22

Table 2: Dice's value for lexical set annotation for the Baseline Algorithm and the LEA Algorithm.

6 Error Analysis

The results presented in the first part of Table 1 were manually inspected to identify sources of errors. In particular, we have noticed that many inaccuracies are due to the automatic mapping of STs to WordNet synsets. For instance, both algorithms failed to recognize *casa* (Eng.: house), corresponding to the ST [[Building]] which was automatically mapped onto *building#n*; they would have succeeded, had the ST been mapped to the more general *construction#n*.

Even when the automatic mapping works, the different structure of the two resources can lead to wrong results. For instance, vehicles such as *elicottero* (Eng.: helicopter) are frequently generalized by the ST [[Vehicle]] in T-PAS and are hyponyms of *vehicle#n* in MWN. However, while in T-PAS [[Machine]] is a hypernym of [[Vehicle]], the same is not true for *machine#n* in MWN. As a consequence, in the sentences in which vehicles are considered members of the lexical set correspondent to [[Machine]], even traversing the MWN hierarchy, the algorithms can not consider these items as valid candidates for the ST [[Machine]].

To solve at least some of these problems, we manually inspected the 40 STs of the sentences of the gold standard, and modified the automatic mapping of 11 of those; for example, we chose to translate the ST [[Building]] to *construction#n*, and mapped [[Machine]] to both *transport#n* and *machine#n*. This led to a significant improvement of the recall for both algorithms, and a minor improvement of the precision, as shown in Table 1.

This improvement is also reflected on the second part of the task (i.e. the creation of the lexical

set). For example, the Dice value for *Crollare#1-SBJ-[[Building]]* improves from 0.4 to 0.71 for the Baseline and from 0.25 to 0.6 for LEA.

Another significant aspect concerns the recognition of proper names: out of the 185 tokens that are -or are part of- proper nouns (137 are related to persons, locations or organizations), the Baseline recognized correctly only 10 (mainly common nouns that are used as proper names), while the LEA algorithm only 26.

Finally, some errors are introduced in the PoS tagging and dependency parsing steps. During the former, an incorrect tag can be assigned to a word (e.g. a noun could be mis-tagged as an adjective) and hinder both algorithms, as the word would not be checked in MWN. The latter only undermines the recall of the LEA algorithm instead. Moreover, LEA does not deal with complex syntactic structure yet (e.g. when our verb is in an infinitive phrase, which is the object of a main verb, such as "[..] e il presidente chiede agli italiani di *ipotecare* la casa [..]", Eng.: [..] and the president asks Italians to *mortgage* their houses [..]).

7 Conclusion and Further Work

In this paper we have presented an experiment for the automatic building of lexical sets for argument positions of the Italian verbs in the T-PAS resource. The method is based on the use of MWN in order to match the STs with the potential fillers of each argument position.

The experiment suggests that LEA can be used to automatically populate the lexical sets with good precision. We believe that significantly better results could be obtained with an accurate manual mapping of the STs to synsets, possibly narrowed to specific senses (e.g. mapping [[Building]] to just the third sense of *construction#n*). Furthermore, recognizing proper nouns proved a difficult task, and even using named entities recognition in addition to MWN was not enough. Therefore a resource to map these nouns to a synset in the WordNet hierarchy is needed; BabelNet (Navigli and Ponzetto, 2012) could prove useful in this sense.

Further work includes the extension of the sentence annotation and lexical set population for all T-PAS and the comparison of the same ST in different T-PASs in order to study Italian verbs' selectional preferences from the perspective of verb selectional classes (for example, all verbs that select [[Food]] as object).

References

Marco Baroni and Adam Kilgarriff. 2006. Large linguistically-processed web corpora for multiple languages. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, pages 87–90.

Patrick Hanks and James Pustejovsky. 2005. A pattern dictionary for natural language processing. *Revue française de linguistique appliquée*, 10(2):63–82.

Patrick Hanks. 1996. Contextual dependencies and lexical sets. *The International Journal of Corpus Linguistics*, 1(1).

Patrick Hanks. 2004. Corpus pattern analysis. In *Proceedings of the 11th EURALEX International Congress, Lorient, France, Universite de Bretagne-Sud*, volume 1, pages 87–98.

Elisabetta Jezek and Francesca Frontini. 2010. From Pattern Dictionary to Patternbank. In G.M. De Schrijver, editor, *A Way with Words: Recent Advances in Lexical Theory and Analysis*, pages 215–239. Kampala:Menha Publishers.

Elisabetta Jezek, Bernardo Magnini, Anna Feltracco, Alessia Bianchini, and Octavian Popescu. 2014. T-PAS: a resource of corpus-derived Types Predicate-Argument Structures for linguistic analysis and semantic processing. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. MultiWordNet: developing an aligned multilingual database. In *Proceedings of the 1st international conference on global WordNet*, volume 152, pages 55–63.

Emanuele Pianta, Christian Girardi, and Roberto Zanoli. 2008. The TextPro Tool Suite. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.

Francesco Sabatini and Vittorio Coletti. 2007. *Dizionario della lingua italiana 2008*. Milano: Rizzoli Larousse.

CJ van Rijsbergen. 1979. *Information Retrieval. 1979*. Butterworth.